Business Process Modelling based on Petri nets

Jianglong Qin\textsuperscript{1,2}, Na Zhao\textsuperscript{1,2}, Zhongwen Xie\textsuperscript{1,2} and Qi Mo\textsuperscript{1,2}
\textsuperscript{1}School of Software Yunnan University Kunming, China
\textsuperscript{2}Key Laboratory in Software Engineering of Yunnan Province Kunming, China
e-mail: king8414@163.com, zhaonayx@126.com, xiezsw56@126.com, moqiyueyang@163.com

Abstract. Business process modelling is the way business processes are expressed. Business process modelling is the foundation of business process analysis, reengineering, reorganization and optimization. It can not only help enterprises to achieve internal information system integration and reuse, but also help enterprises to achieve with the external collaboration. Based on the prototype Petri net, this paper adds time and cost factors to form an extended generalized stochastic Petri net. It is a formal description of the business process. The semi- formalized business process modelling algorithm based on Petri nets is proposed. Finally, The case from a logistics company proved that the modelling algorithm is correct and effective.

Keywords—Business Process Modelling; Basic Block; Modelling Algorithm; Petri nets

1 INTRODUCTION

Business process everywhere, to improve the business process management and processing capabilities has very important significance to improve work efficiency for the government and to improve business performance for the enterprise. With the development of information technology, it creates the technical foundation for a substantial increase for business process management capabilities and processing power [1].

Business process modelling is the way business processes are expressed. Business process modelling is the foundation of business process analysis, reengineering, reorganization and optimization. The common modelling languages include UML[2], EPC[3], Workflow Graph[4], BPMN[5], Pi-Calculus[6] and Pet nets[7]. Petri is not only intuitive, but also a formal definition. Based on the prototype Petri net, this paper adds time and cost factors to form an extended generalized stochastic Petri net. It is a formal description of the business process. The semi- formalized business process modelling algorithm based on Petri nets is proposed.

2 DEFINITION OF BUSINESS PROCESS

Definition 2.1[7] A triple $N=\langle C, A, F \rangle$ is called a net if

$(1) C \cup A \neq \Phi$;

$(2) C \cap A = \Phi$;

$(3) F \subseteq (C \times A) \cup (A \times C)$ is a binary relation, the flow relation of $N$;

$(4) \text{dom}(F) \cup \text{cod}(F) = C \cup A$.

C which is called the condition set of $N$, $A$ is called the activity set, $F$ is called arc set. $X = C \cup A$ is called the $N$ element set. Elements in $C$ are called the condition or element $C$, Elements in $A$ are called activity or element $A$. The $\Phi$ is empty. $x$ is the two-set (space) of the Cartesian product operation, so $F$ is the set of ordered pair composed of a $C$ element and a $A$ element. $\text{dom}(F)$ is the order dual of $F$ contains the first element (starting) of the set, $\text{cod}(F)$ is the second element (the end) of the collection: $\text{dom}(F) = \{x | y: (x, y) \in F\}; \text{cod}(F) = \{x | y: (y, x) \in F\}$. Net standard graphics is that circle represents the condition, with the box that transition s from $x$ to $y$ by a directed arc (arrow) that ordered dual (x, y). (hence (x, y) $\in F$), it is also known as directed arcs.

Definition 2.2 Satisfy the following six conditions set $\Sigma = (C, A, F, M, \lambda, g)$ is called a business process system:

$(1) (C, A, F)$ is a net;

$(2)$ where $T$ is divided into time transition set $At$ and instantaneous transition set $Ai$ sub-sets. The implementation of the time transition will take some time, the instantaneous transition implementation time is negligible, that is, $A=At \cup Ai$, $At \cap Ai = \Phi$;

$(3)$ $F$ in the permitted inhibitor arcs, inhibitor arcs exist only from the position to transition the arc. Inhibitor arcs can be implemented by the conditions of the original link becomes unenforceable (disable) the conditions, the original non-implementation of the conditions can be implemented into the conditions, and transition in the linked implementation, not marked out from the connected position;
(4) \( \lambda = \{ \lambda_1, \lambda_2, \ldots, \lambda_m \} \), \( \lambda_i \) is the average implementation rate of the transition \( ai (ai \in A) \), which represents average implementation times in the case of the implementation in unit time. The unit is times/hour.

(5) \( g = \{g_1, g_2, \ldots, g_m\} \), \( g_i \) is the average implementation costs of the transition \( ai (ai \in A) \), which represents implemented cost in the case of implementation per unit time. The unit is the yuan/hour.

Where each delay of the transition \( 1/\lambda = \{1/\lambda_1, 1/\lambda_2, \ldots, 1/\lambda_m \} \) are exponentially distributed, the cost of each transition \( g = \{g_1, g_2, \ldots, g_m\} \) are exponentially distributed.

**Definition 2.3** Let \( \Sigma = (C, A, F, M, g) \) represents a business process system. Let \( M_0 \in M \) (\( M_0 \subseteq C \)), \( p = (C, A, F, M_0, g) \). \( M_0 \) is called the initial marking of \( p \) and \( p \) is called a business process. If confusing, then, \( C, A, F, M_0, g \) respectively \( p.C, p.A, p.F, p.M_0, p.g \) to represent.

**Definition 2.4** Let \( p = (C, A, F, M_0, g) \) represents a business process. For \( p \) in any one activity \( a \in A \), \( \text{Inflow} (a) = \{(x, a) | (x, a) \in F, x \in C \} \), \( \text{Inflow} (a) \) is called the input stream for the activity of \( a \), \( \text{Outflow} (a) = \{(a, y) | (a, y) \in F, y \in C \} \), \( \text{Outflow} (a) \) is called the output stream for the activity of \( a \).

**Definition 2.5** Let \( N = \{C, A, F\} \) as a net. For \( x \in C \cup A \), \( .X = \{y | y \cup A \cap (y, x) \in F \} \), \( .X \) is called the front set or the input set of \( x \), \( \_X = \{y | y \cup A \cap (x, y) \in F \} \), \( \_X \) is called the back set or the input set of \( x \).

**Definition 2.6** Let \( N = \{C, A, F\} \) as a net, \( \forall c1 \in C, \forall c2 \in C, c1 \neq c2, \text{if and only if} c1 \cap c2 \neq \Phi \vee c1 \cap c2 = \Phi, c1 \text{ and } c2 \text{ are called the adjacent condition, denoted by } \text{border}(c1, c2). \text{Otherwise } c1 \text{ and } c2 \text{ are called non-adjacent condition, denoted by } \neg \text{border}(c1, c2). \)

**Definition 2.7** Let \( N = \{C, A, F\} \) as a net, \( \text{StartPlace} (N) = \{c | a \in A : (a, c) \not\in F \} \), \( \text{EndPlace} (N) = \{c | a \in A : (c, a) \not\in F \} \), \( \text{EndPlace} (N) \) is called termination condition of net; \( \text{SpecialPlace} (N) = \text{StartPlace} (N) \cup \text{EndPlace} (N) \), \( \text{SpecialPlace} (N) \) is called special condition of the net.

**Definition 2.8** Sequence block, concurrency block, selection block and iteration block are called basic block. A basic block can be described as a 5-tuple \( b = (C, A, F, Ae, Ax) \) where

1. \( C, A \) and \( F \) are called the condition set, the activity set and the arc set respectively.
2. \( Ae, Ax \subseteq A \) are called the entrance and the exit of \( b \) respectively.

Basic blocks, which are enclosed by dotted lines in figures, are described as follows

Where each \( ei (i=1,2,\ldots) \) denotes an activity.
3 BUSINESS PROCESS MODELLING

Business process modelling method is from top to bottom, and gradually decomposed. The first step in any business process is to create an initial block and the definition of the initial block is shown in Figure 5. Business process by layer refinement, the modeler can get a business process to meet their needs model. The modelling of business processes is described in Function 3.1. Two sub-functions are called in the process of business process modelling. Function 3.2 is used for business process layer by layer refinement. Function 3.3 is used to establish a link between non-adjacent libraries in a business process.

Function 3.1 (Business process modelling method)

```c
p Modelling_Business_Process( )
{
    /* I represents the number of business processes */
    int i;
    /* Max represents the maximum number of stages */
    p(0), ..., p(MAX):
    Define p(0) as an initial block;
    If (The initial identity of the initial block must be determined by other conditions)
    {
        /* p(0).M0 represents the initial identity of the business process p(0) */
        p(0).M0=Φ;
    }
    else
    {
        p(0).M0={start};
    }
    i=0;
    /* Call Function 3.2*/
    p(0)=decompose(i);
    /* Call Function 3.3*/
    p(0)=Modelling_NoBorderCondition_Relation( p(0) );
    {\lambda : p(0).A_i \rightarrow \lambda_i
    g : p(0).A_i \rightarrow g_i
    return p(0);
    }
    Function 3.2
    p decompose(int i)
    {
        p(i+1)=p(i);
        for(a : p(i).A)
        {
            Analyze activity a;
            if (Activity a needs to be broken down into basic blocks)
            {
                Build a basic block b=\{C, A, F, A_e, A_x\}
                p(i+1).C= p(i+1).C \cup b.C;
                p(i+1).A= p(i+1).A - \{a\} \cup b.A;
                p(i+1).F= p(i+1).F - Inflow(a) - Outflow(a) \cup b.F
                \cup \{(x, y) | (x, a) \in Inflow(a) \land y \in b.A_e \}
                \cup \{(x, y) | x \in b.A_x \land (a, y) \in Outflow(a)\};
            }
        }
        If (p(i+1) need to continue refinement refinement)
        {
            decompose(i+1);
        }
        else
        {
            return p(i+1);
        }
    }
    ```

Figure 4. Iteration Block

Figure 5. Initial block
Function 3.3

```c
p Modelling_NOBorderCondition(p p1)
{
    p p2;
    p2=p1;
    for(c1 ; p1.C; c1 = SpecialPlace(N))
    {
        for(c2 ;p1.C, c2 = SpecialPlace(N))
        {
            if( ¬border(c1,c2) && There is some basic block between c1 and c2)
            {
                Build a basic block d={C, A, F, Ae , Ax};
                if( (C1 is connected to d.Ae) && (C2 is connected to d.Ax) )
                {
                    p2.F= p1.F U d.F U {( c1,y)| y ∈ b.Ae}
                     {( y, c2)| y ∈ b.Ax};
                }
                else if( (C1 is connected to d.Ax ) && (C2 is connected to d.Ae) )
                {
                    p2.F= p1.F U d.F U {( c2,y)| y ∈ b.Ae}
                     {( y, c1)| y ∈ b.Ax};
                }
            }
        }
    }
    return p2;
}
```

### 4 CASE STUDY

This paper combines the application examples of an enterprise to illustrate the modelling of extended generalized stochastic Petri nets. Fig.6 is a logistics company's operating process.

![Figure 6. The flow of a logistics company](image)

According to the business process modelling algorithm(Function 3.1), get a formal description of business process modelling. The description is as follow.

- \( A = \{a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12,a13,a14,a15,a16,a17,a18,a19,a20,a21,a22,a23\} \)
- \( A_i = \{a6,a9,a11\} \)
- \( A_t = \{a1,a2,a3,a4,a5,a7,a8, a10, \)
a12,a13,a14,a15,a16,a17,a18,a19,a20,a21,a22,a23};

C={c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,
c11,c12,c13,c14,c15,c16,c17,c18,c19};

F={(c1,a1),(a1,c2),(c2,a2),(c2,a3),(a2,c3),(a4,c3),(a5,c4),(a4,c5),(a6,c5),(a6,c6),(c5,a7),(c6,a8),(a7,c7),(a8,c8),(c7,a9),(a9,c9),(c9,a10),(a10,c10),(c10,a11),(a11,c11),(c11,a12),(a12,c12),(c12,a13),(a13,c13),(c13,a14),(a14,c11),(c13,a15),(a15,c14),(c14,a16),(a16,c15),(c15,a17),(a17,c10),(c14,a18),(a18,c18),(c9,a19),(a19,c16),(c16,a20),(a20,c18),(c9,a21),(a21,c17),(c17,a22),(a22,c18),(c18,a23),(a23,c19)};

M0={c1};

Finally, we get the extended generalized stochastic Petri nets representation of business processes. It shown in Figure 7. According to Figure 7, we can quantify and optimize business processes of the logistics company.

5 CONCLUSIONS

In this paper, the concept of cost is introduced to the generalized stochastic Petri nets. It is called it extended generalized stochastic Petri nets. At the same time, basic block is introduced to business process modelling. Business process is got described with the extended stochastic Petri net. The semi-formalized business process modelling algorithm based on Petri nets is proposed. Finally, the case from a logistics company proved that the modelling algorithm is correct and effective.

Acknowledgment

This work has been supported by the National Science Foundation of China under Grant No. 61379032.

References